Discussion 5: Algorithmic Complexity

Algorithmic Complexity: Definitions

1. What is runtime? How do we measure it?

_____

_____

2. If a function runs in O(n) time, that means it runs…

    ○ in linear time at worst   ○ in linear time on average   ○ in linear time at best
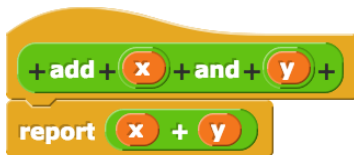
Understanding Runtimes

1. Fill in the following chart:

Runtime	Notation	As input size increases by…	The number of steps change by…
Constant		+1	
Logarithmic (base B)		xB	
Linear		+1	
Quadratic		x2	
Exponential (base B)		+1	

Runtime Practice

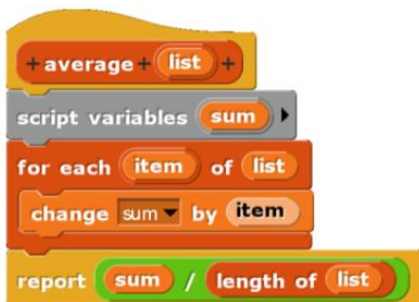1. Find the runtime of each of the following blocks or processes:

a.

Runtime: _____

b. This block takes in a value and a list and searches through every item in the list one by one to see if it can find that value.

Runtime: _____

c.

Runtime: _____

d. This block takes in a value and a sorted list and searches for the value in the sorted list. Every iteration of the algorithm, it figures out which half of the list the value would be in, and then only searches in that half of the list.

Runtime: _____

e.

```
+ (num) + mod + 7 +
  script variables (num copy) ▶
  set [num copy ▼] to (num)
  repeat until ((num copy) mod (7) = 0)
    change [num copy ▼] by (1)
  report ((num copy) - (num))
```

Runtime: _____

f. You know a secret, and you want to share it with the world. In *state 0,* you are the only person who knows the secret. Then in *state 1*, you share the secret with two friends, so three total people know the secret. Then in *state 2*, both of your friends tell two of their friends, so seven total people know the secret. This pattern continues indefinitely. As a function of the *state*, what is the order of growth of the number of people who know the secret?

Runtime: _____

f.

```
+ mystery + (n) +
  script variables (x) ▶
  set [x ▼] to (1)
  repeat until ((x) > (n))
    change [x ▼] by (x)
  report (x)
```

Runtime: _____

g. This block finds the number of pairs, sets of two values that are equal, in a list. It first starts out by finding pairs for the first element: it searches through everything after the first element and counts the number of values equal to the first element. Then, it finds pairs for the second element by searching through everything after the second element for pairs. It continues with this process until it reaches the end of the list.

Runtime: _____

h.

```
+ sort + (list :) +
  script variables (sorted list) (min) (min_index) ◀▶
  set [sorted list ▼] to (list ▶)
  repeat until ((length of (list)) = 0)
    set [min_index ▼] to (1)
    set [min ▼] to (item (1 ▼) of (list))
    for (i) = (1) to (length of (list))
      if ((item (i) of (list)) < (min))
        set [min ▼] to (item (i) of (list))
        set [min_index ▼] to (i)
    add (item (min_index) of (list)) to (sorted list)
    delete (min_index) of (list)
  report (sorted list)
```

Runtime: _____